

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/220427764>

Symbolic Modeling in Practice – Introduction.

Article in Communications of the ACM · January 1999

DOI: 10.1145/291469.293164 · Source: DBLP

CITATIONS

10

READS

434

2 authors:



Hermann Kaindl

TU Wien

303 PUBLICATIONS 2,795 CITATIONS

[SEE PROFILE](#)



John M. Carroll

Pennsylvania State University

820 PUBLICATIONS 32,974 CITATIONS

[SEE PROFILE](#)

SYMBOLIC MODELING *in* PRACTICE

MODELING IS A COMPLEX ACTIVITY OF ABSTRACTING INFORMATION AND KNOWLEDGE FROM A particular domain in order to achieve a model containing the essentials from the perspective of the modelers and their given goals. While this definition includes building numeric models, for instance, we will focus here on symbolic modeling.

Symbolic modeling is employed in a variety of fields. Examples include task and domain modeling in human-computer interaction (HCI), semantic data modeling for database systems, knowledge acquisition and representation in artificial intelligence (AI), and object-oriented (OO) modeling in software engineering.

Industrial practitioners may be confused about the latest modeling trends. For instance, when object-oriented approaches became suddenly popular, the close relations to entity-relationship modeling (from the database community) and to frame systems and semantic networks (from AI) were not clear to all computer professionals even if they have applied one or the other of these approaches previously.

Another important issue for industrial practitioners is judging whether or not a new approach is useful for them or how to combine it with previous approaches. An example is the increasing popularity of interaction scenarios in several areas, whose application may seem to be incompatible with functional specifications.

From a theoretical perspective, it is well understood that these modeling efforts somehow overlap, but it is not well understood exactly how they relate to one another: mutual recognition of corresponding work is frequently inadequate because similar work is often pursued concurrently in different fields under different names.

We think the fundamental ideas in symbolic modeling should be broadly clarified to further a common understanding. We attempted to improve the situation in past publications¹; however, none of the previous publications has the scope of both technologies and fields we believe is important to cover now, nor the common coverage of fields that is important for practice. In presenting the articles in this section we intend to improve the practice of symbolic modeling by increasing awareness of the

¹The July/August 1994 issue of the *International Journal of Human-Computer Studies (IJHCS)* [2], includes papers illustrating mutual influences among the fields of OO development, AI, and HCI. (That special issue was edited by the first guest editor of this special section, who also tried to raise the awareness of such commonalities inside the OO community [3]). *Scenario-Based Design: Envisioning Work and Technology in System Development* [1], contains a series of contributions illustrating the role of the user-task scenarios in providing a framework for OO software design and HCI.

HERMANN KAINDL AND JOHN M. CARROLL, GUEST EDITORS

previously mentioned issues in practice, by providing some pointers to work that crosses the boundaries, and by covering a variety of approaches.

We have assembled an excellent collection of authors to present a set of perspectives on new issues and approaches to symbolic modeling. We particularly wanted to bridge and integrate concepts and methods in symbolic modeling across disciplines. In fact, each article in this special section integrates issues related to symbolic modeling in two different fields.

Still, it is interesting to observe that one topic is shared by three of the four feature articles: OO modeling. At the current state of practice, OO approaches seem to be having a strong impact on software development. Some of the major OO analysis and design methods also led to the definition of the so-called Unified Modeling Language (UML, see www.rational.com/uml/). We believe the availability of such a standardized diagrammatic language may be important for the practice of symbolic modeling.

However, establishing such a standard carries the danger of freezing parts of these particular methods, and impeding the use of other modeling views and techniques. In fact, the very idea of OO modeling is much broader than what is covered by the UML. Evidence for this claim can be found in the special issue of the IJHCS (which centered around OO approaches in AI and HCI) mentioned in the footnote to this introduction, as well as in several of the articles in this special section. In particular, the article by Mylopoulos, Chung, and Yu provides an authoritative review of conceptual modeling (although this is only one aspect of their article), which attempts to clarify the history of OO modeling. So a shared language such as the UML may help, but it does not provide a comprehensive framework for all the relevant issues of symbolic modeling.

The three articles in this special section related to OO modeling suggest various ways of broadening it:

- Mylopoulos, Chung, and Yu sketch a rather positive view of OO methods, but more importantly they address the issue of nonfunctional requirements (also called quality or constraint requirements) in their approach to “satisficing” what they call softgoals. Nonfunctional requirements are generally insufficiently studied, and the current meth-

ods of OO analysis and design are particularly ignorant in this regard. This article represents an important contribution by addressing this issue in the context of OO modeling.

- Butler, Esposito, and Hebron apply the use-case notation of the UML, but they also introduce more rigorous task modeling in their approach. The integration of the two demonstrates how business-process modeling can converge with the modeling of software. The authors suggest it is necessary to use HCI approaches in order to make a software application more usage-centered.
- Rosson presents an approach that integrates the development of task models and OO models through scenario-based usability rationale. This article focuses on how modeling in HCI and OO can be linked together.

While Butler, Esposito, and Hebron make use of a scenario-based approach for software modeling only, this integration by Rosson is primarily based on interaction scenarios. Note that such scenarios have a wider scope than those defined in the UML (see, for example, [1]). In fact, scenario approaches have been widely developed—in HCI and in requirements engineering, for example. We consider this a particularly healthy sign of the underlying ideas. In that interdisciplinary spirit, we have included a brief review by Jarke of current issues in scenario-based development.

While Mylopoulos, Chung, and Yu elaborate on softgoals, the goals to be achieved by users in the course of applying an interaction scenario are typically left implicit (and they are also not covered, for example, by the UML). Being aware of the goal(s) of some sequence of (inter)actions, however, facilitates their understanding. We argue that they should be made explicit [2] and should be represented together with their relationship to the corresponding scenarios [5].

The recent scenario approaches may seem to be in conflict with more traditional functional specifications, which appear not to be very popular these days in the spirit of the current OO view. However, it is already recognized that both approaches are needed and should complement each other [1], and a concrete approach to combine them can be found in [4]. The more recent combination of scenarios both with functions and with goals avoids the possibility that functional specifications may become divorced from users' needs, and is the basis for a defined design process [5].

One of the strong aspects of diagrammatic languages such as the UML is their support of visual

modeling. While it is not a general truth that visual languages must be better than others, it seems that practitioners like visual modeling. In this spirit, Gaines and Shaw present their approach of a visual language for knowledge modeling. It is based on AI technology of automated reasoning using symbolic models that are formally represented and operational. In its implementation, automatic inferences are drawn by machine from models represented in this language, for example, showing the effects of business rules in a company model.

Even more importantly, however, Gaines and Shaw also present their approach for embedding this visual language in hypertext documents. In effect, this approach makes the hypertext documents *active documents*. While the embedding of active text in hypertext in the sense of OO message passing has been treated previously [6], automatic inferences in hypertext are certainly a very fruitful combination of AI and HCI technology. For the practical use of this approach, its availability for Web pages is especially important.

Symbolic modeling is an enduring, pervasive topic that should be considered an "evergreen" issue. While more trendy topics come and go quite rapidly, symbolic modeling has been around for a long time and we conjecture that it will maintain (or increase) its importance over time. For now, the main purpose of this section is to improve the practice of symbolic modeling—it is our hope the articles presented here will assist in doing so. □

REFERENCES

1. Carroll, J.M., Ed. *Scenario-Based Design: Envisioning Work and Technology in System Development*. Wiley, NY, 1995.
2. Carroll, J.M., Mack, R.L., Robertson, S.P., and Rosson, M.B. Binding objects to scenarios of use. *International J. of Human-Computer Studies* 41, 1/2, (July/Aug. 1994), 243–276.
3. Kaindl, H. Object-oriented approaches in software engineering and artificial intelligence. *J. of Object-Oriented Programming* 6, 8 (Jan. 1994), 38–45.
4. Kaindl, H. An integration of scenarios with their purposes in task modeling. In *Proceedings of the Symposium on Designing Interactive Systems: Processes, Practices, Methods, and Techniques (DIS '95)* (Ann Arbor, MI, Aug. 1995), pp. 227–235.
5. Kaindl, H. Combining goals and functional requirements in a scenario-based design process. In *People and Computers XIII, Proceedings of Human Computer Interaction '98 (HCI '98)* (Sheffield, UK, Sept. 1998), pp. 101–121.
6. Kaindl, H. and Snaprud, M. Hypertext and structured object representation: A unifying view. In *Proceedings of the Third ACM Conference on Hypertext (Hypertext '91)* (San Antonio, TX, Dec. 1991), pp. 345–358.

HERMANN KAINDL (hermann.kaindl@siemens.at) is a senior consultant at Siemens AG Österreich in Vienna, Austria.

JOHN M. CARROLL (carroll@cs.vt.edu) is a professor of Computer Science, Education, and Psychology, and the director of the Center for Human-Computer Interaction at Virginia Tech in Blacksburg, Virginia.